AD-A100 457 TEXAS UNIV AT AUSTIN CENTER FOR CYBERNETIC STUDIES F/G 12
NETGEN-II: A SYSTEM FOR GENERATING STRUCTURED NETWORK-BASED MAT--E
JAN 81 J ELAM, D KLINGMAN N00014-80-C-0242
UNCLASSIFIED CCS-393

1 OF 1
AD
A100457

END
DATE
FILMED
7-81
DTIC

AD A100457

LEVEL II

12

CENTER FOR CYBERNETIC STUDIES

II 12

CCS - 393

NETGEN-II: A SYSTEM FOR GENERATING STRUCTURED NETWORK-BASED MATHEMATICAL PROGRAMMING TEST PROBLEMS

by

Joyce Elam*

Darwin Klingman**

Jan[redacted]981

14

* Visiting Associate Professor of Statistics and Operations Research, Department of General Business, BEB 600, The University of Texas, Austin, Texas 78712

** Professor of Operations Research and Computer Sciences, Department of General Business, BEB 600, The University of Texas, Austin, Texas 78712

This research was supported in part by the National Science Foundation Contract MCS-7827590 and by Office of Naval Research Contract N00014-80-C-0242. Reproduction in whole or in part is permitted for any purpose of the U.S. Government.

CENTER FOR CYBERNETIC STUDIES

A. Charnes, Director
BEB 203E
The University of Texas at Austin
Austin, Texas 78712
(512) 471-1821

ABSTRACT

The increased importance of designing and implementing algorithms to solve particular management problems has created the need for more robust test problem generators that can match the overall structure and parameter values of these problems. Of particular interest are management problems that can be modeled using a network structure. This paper discusses the design of a system for generating network-based mathematical programming test problems that conform to user-supplied structural and parameter characteristics.

1. INTRODUCTION

Recent years have seen an increase in the development of efficient algorithms for solving various classes of mathematical programming problems. This development has been motivated by a desire to reduce solution time and computer costs in solving current problems and/or to solve problems that are computationally infeasible using existing methods. The efficiency of an algorithm is based upon several criterial including its effectiveness with respect to different problem classes, its speed, capacity, and accuracy. Since existing theory alone cannot provide measurements for these criteria, emperical computational testing must be employed. A necessary prerequisite for such testing is the ability to construct and/or obtain test problems with known optimal solutions. The literature contains several sets of randomly-generated test problems have used for this purpose [3, 11, 12, 13].

One class of mathematical programming problems that has received extensive interest in recent years can be broadly defined as network and network-related problems. Pure network problems represent a special class of linear programming problems and embody a group of distinct model types: shortest path, assignment, transportation, and transshipment. Generalized network problems represent a broader classification of linear network-related problems. Other network-related problems include linear programming problems that have a network substructure such as multi-commodity networks, a pure or generalized network with extra constraints, or even a linear programming problem with GUB constraints.

The development of efficient solution methods and new modeling techniques for expressing problems in a pictorial network formulation [1,2,6,7,8,9,10] has led to the increased use of network-based models in government and business. These models range from rather straightforward network applications such as production planning and distribution to less obvious applications involving the refueling of nuclear reactors and optimal lot sizing and machine loading for multiple products. As network model-based systems are designed and implemented to handle larger and more diverse types of network and network-related problems, it is highly desirable to have the capability to generate test problems that match the overall structure and parameter value ranges of the models the systems are being developed to solve.

NETGEN [12], currently the most widely used generator for network test problems, can only generate pure network problems. In addition, NETGEN is limited in its ability to capture the characteristics of real-world problems in the network problems it can generate. For example, NETGEN cannot generate multi-period transportation or transshipment problems. The increased emphasis on modeling and the development of computer-based decision support systems built around networks models and employing network algorithms have created a need for a more robust and powerful test problem generator that is driven by user-supplied problem characteristics. This paper describes the design of NETGEN-II, a system developed in response to this need.

2. NETGEN-II OVERVIEW

A distinguishing feature of NETGEN-II is the use of a model specification language for describing the structural characteristics of a

model and the parameters values to be used in generating a problem from this model structure. A user can create a model specification with this language either through the interactive builder component of NETGEN-II or directly through a system-supplied editor. In addition, NETGEN-II provides access to a library of "standard" network model structures that can be used as a basis for creating a model specification. Once a model specification has been created, NETGEN-II provides the capability to randomly generate a family of problems from this specification, where each problem has the same underlying structure and parameter value ranges. NETGEN-II represents each generated problem in MPSX input format. The overall architecture of NETGEN-II is shown in Figure 1.

The remainder of this paper presents a brief overview of the model specification language. A more complete description of the language and the NETGEN-II is contained in [4,5].

2. MODEL SPECIFICATION LANGUAGE

The design of the model specification language is built around the concept of a "template" as the vehicle for generating the network structure of a mathematical programming test problem. An example template definition using the model specification language and one possible template that could be generated from this definition is shown in Figure 2. A template is defined through four types of statements: NODE CLASS, SIZE, RELATIONSHIP, and CONNECT. The NODE CLASS and RELATIONSHIP statements define the types of nodes (called classes) and the types of linkages between these node classes (called relationships) that are to be represented in the basic graph structure of the template. A relationship always involves two node classes and is directed from the first node class to the second node

Architecture of

NETGEN-II

System

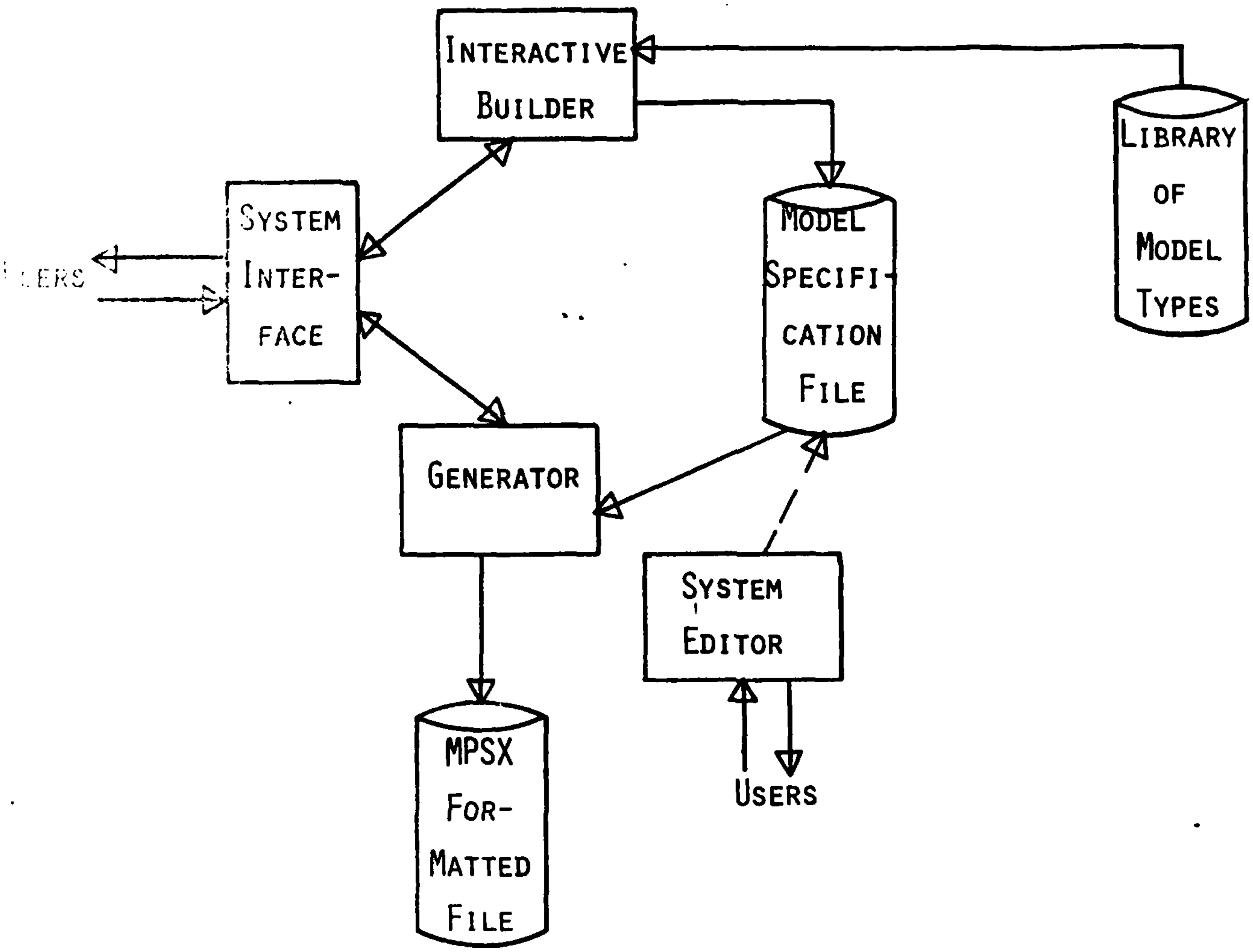

Figure 1

class. The SIZE and CONNECT statements specify how the template is to be created using the node classes and relationships. The SIZE statements define the number of nodes in each class, where the nodes in a class are assumed to form an ordered set that is numbered consecutively beginning at 1. The CONNECT statements specify how to generate the arcs that form each relationship. For a given relationship, a CONNECT statement identifies a subset of nodes in the from node class and in the to node class of the relationship that are to be connected. The syntax defined for the CONNECT statement allows several different ways of identifying a subset. The nodes in a subset can be explicitly defined--i.e., all nodes in a class (all plant) or all nodes between the mth and nth node in a class (ordered set 1-2 of warehs); or the identification of nodes to be contained in a subset can be deferred until the template is generated--i.e., a given number of nodes chosen randomly from a class (random set of 3 cust). When a CONNECT statement is processed during template generation, an arc is created from <u>each</u> node contained in the to node subset to <u>every</u> node contained in the from node subset.

After the template is defined, the entire network structure is defined by specifying the number of times the template is to be repeated and the linkages to be used in joining the templates together. Figure 3 illustrates the model specification statements for defining a network using the template defined in Figure 2 and the network problem that would result from these statements. A network is defined through four types of statements: TEMPLATE, NODE CLASS, RELATIONSHIP, CONNECT. The TEMPLATE statement defines the number of repeating templates in the network and assigns a label to each template instance. The NODE CLASS statement is identical in format and meaning to the one defined for the template and is used to

TEMPLATE DEFINITION

```
NODE CLASSES plant, warehs, cust

SIZE IS 2 FOR plant

SIZE IS 3 FOR warehs

SIZE IS 5 FOR cust

RELATIONSHIP ship (plant,warehs)

RELATIONSHIP sell (warehs,cust)

CONNECT IN RELATIONSHIP ship FROM all plant  TO all warehs

CONNECT IN RELATIONSHIP sell FROM ordered set 1-2 of warehs
                                  TO random set of 3 cust

CONNECT IN RELATIONSHIP sell FROM warehs 3 TO random set of 2 cust
```

TEMPLATE INSTANCE

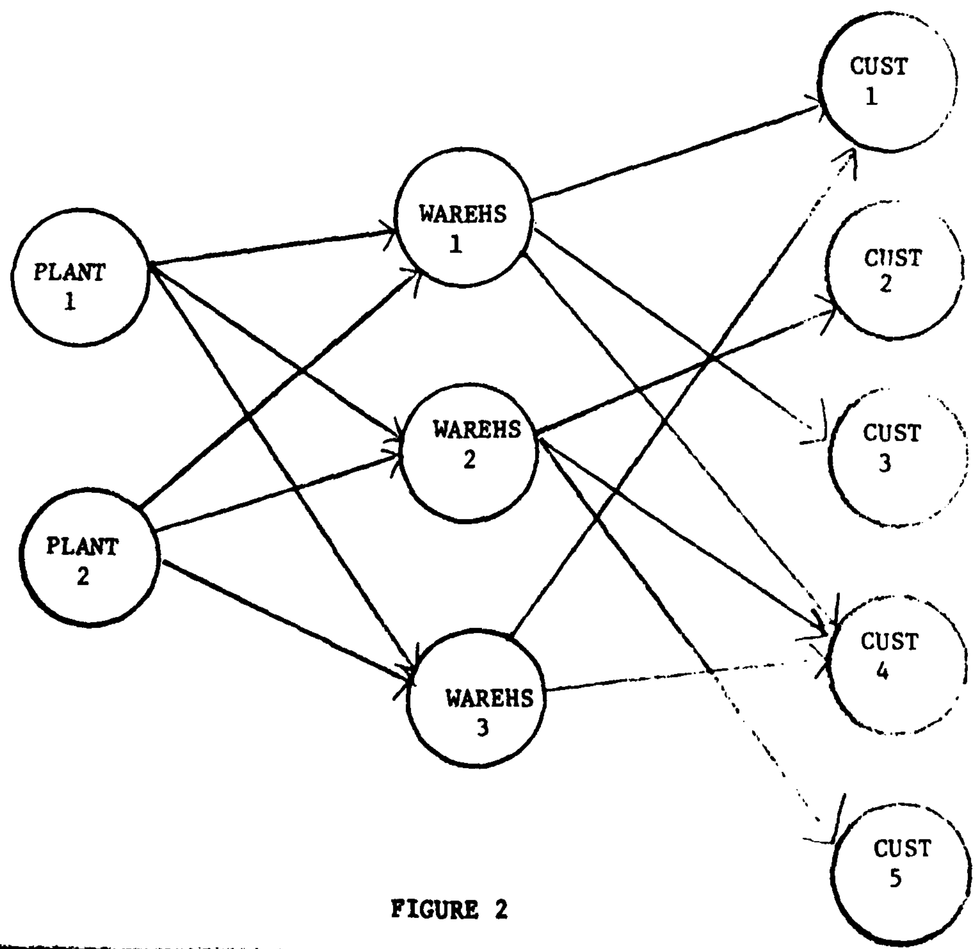

FIGURE 2

define node classes that exist outside the template. (In Figure 3, all node classes exist within a template and thus, there are no NODE CLASS statements required for the network definition.) The RELATIONSHIP and CONNECT statements are identical in format and meaning to the ones defined for the template with the exception that the CONNECT statements must qualify the from and to node class subsets with a template identifier.

In addition to defining the network structure of a problem, the model specification language provides statements for defining any additional constraints that cannot be represented directly in the network structure. For example, the condition that total inventory at the at the end of the first period must not exceed 10,000 in the problem defined in Figures 2 and 3 could be expressed as follows:

```
SUM OF FLOWS IN inventory BETWEEN period1 AND period2
IS LESS THAN 10000
```

The remaining requirement of the model specification language is to define the parameter values that are to be associated with the network structure of the problem. These parameters include supply and demand values for elements of node classes or subclasses and costs, bounds, and multipliers for elements in relationships. Parameter values can be expressed as constants, a range of values, and/or in terms of previously defined parameters. Some example model specification statements for assigning

NETWORK DEFINITION

```
TEMPLATES period1, period2

RELATIONSHIP inventory (warehs,warehs)

CONNECT IN inventory FROM all warehs in period1 TO corresponding
       warehs in period2
```

NETWORK INSTANCE

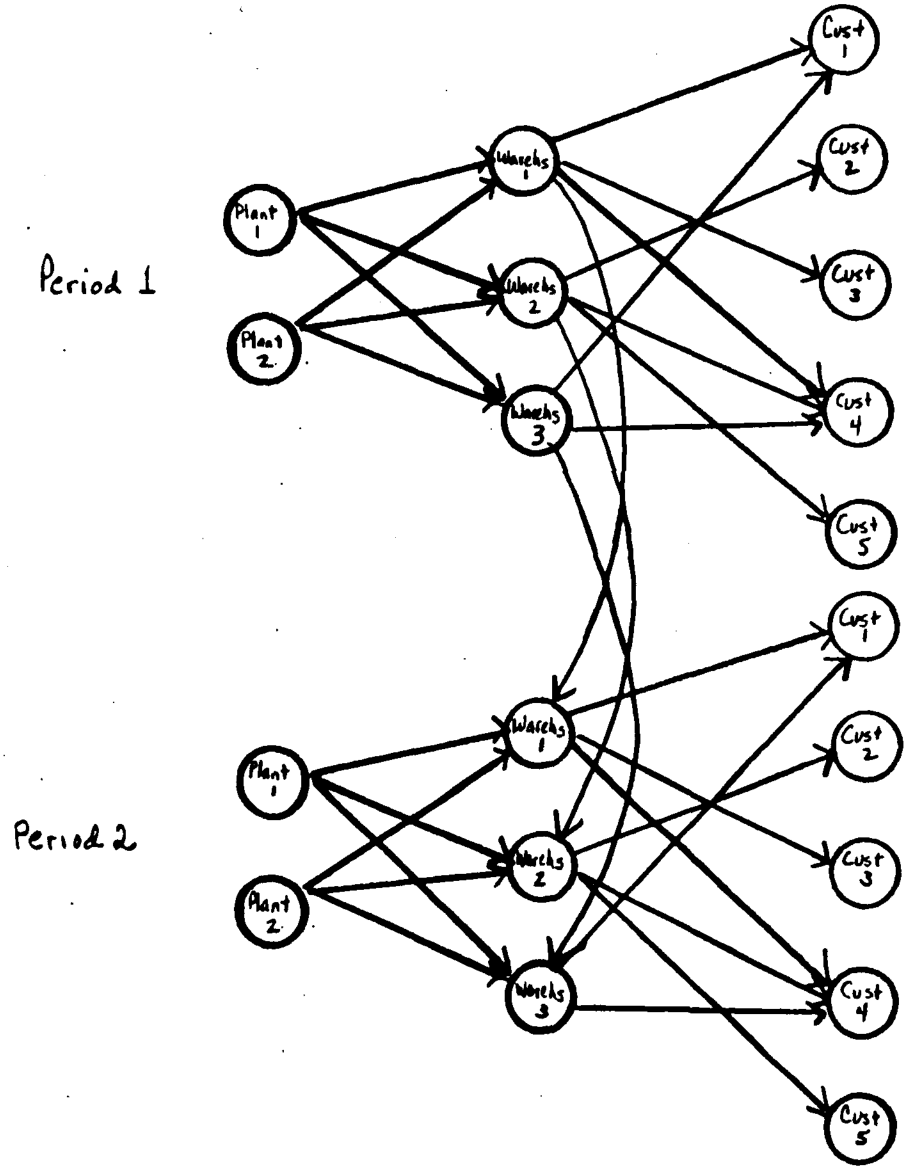

FIGURE 3

parameter values to the problem defined in Figures 2 and 3 are given below:

```
SUPPLY for each plant in period1 IS 500

SUPPLY FOR each plant in period2 IS PREVIOUS SUPPLY * 1.1

DEMAND FOR ordered set 1-3 of cust in each template IS
       random 50,150

COST FOR ship in each template for each arc IS linear
       random 5,15
```

4. CONCLUSIONS

The increased importance of designing and implementing algorithms to solve management problems that can be modeled using network structures has created the need for more robust test problem generators that can match the overall structure and parameter values of these problems. This paper has discussed the overall design of a system that meets this need and the model specification language that is used to define the structural and parameter characteristics to be represented in a test problem.

REFERENCES

1. R. Barr, F. Glover, and D. Klingman, "Enhancements of Spanning Tree Labeling Procedures for Network Optimization,"Research Report CCS 262, Center for Cybernetic Studies, The University of Texas at Austin, (1976).

2. G. Bradley, G. Brown, and G. Graves, "Design and Implementation of Large-Scale Primal Transshipment Algorithms,"Management Science 24, 1 (1977) 1-34.

3. J. C. P. Bus, "A Proposal for the Classification and Documentation of Test Problems in the Field of Nonlinear Programming," Report No. NN 9/77, Stichting Mathematisch Centrum 2 e Boerhaavestraat 49 Amsterdam 1005, HOLLAND, (1977).

4. J. Burruss, J. Elam, and D. Klingman, "NETGEN-II: User's Manual," Research Report, Center for Cybernetic Studies, The University of Texas at Austin, (1980).

5. J. Burruss, J. Elam, and D. Klingman, "The Design of a Generator for Structured Network-Based Problems," Research Report, Center for Cybernetic Studies, The University of Texas at Austin, (1980).

6. A. Geoffrion, "Comments on Mathematical Programming Project Panel on Futures," SHARE XIV Meeting, Los Angeles, (1975).

7. F. Glover, D. Karney, and D. Klingman, "The Augmented Predecessor Index Method for Locating Stepping Stone Paths and Assigning Dual Prices in Distribution Problems," Transportation Science 6,2 (1972) 171-179.

8. F. Glover, D. Karney, and D. Klingman, "Implementation and Computational Study on Start Procedures and Basis Change Criteria for a Primal Network Code," Networks 4,3 (1974) 191-212.

9. F. Glover and D. Klingman, "The Simplex SON Algorithm for LP/Embedded Network Problems," Research Report CCS 317, Center for Cybernetic Studies, The University of Texas at Austin, (1977).

10. F. Glover and J. Mulvey, "Equivalence of the 0-1 Integer Programming Problem to Discrete Generalized and Pure Networks," MSRS 75-19, University of Colorado, Boulder, Colorado (1975).

11. J. Haldi, "25 Integer Programming Test Problems," Working Paper No. 43, Graduate School of Business, Stanford University.

12. D. Klingman, A. Napier, and J. Stutz, "NETGEN: A program for Generating Large-Scale Capacitated Assignment, Transportation, and Minimum Cost Flow Network Problems," Management Science 20, 5 (1974) 814-821.

13. J. Rosen and S. Suzuki, "Construction of Nonlinear Programming Test Problems," Comm. ACM 8, 2 (1965) 113.

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER CCS 393	2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) NETGEN-II: A System for Generating Structured Network-Based Mathematical Programming Test Problems		5. TYPE OF REPORT & PERIOD COVERED 6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s) J. Elam and D. Klingman		8. CONTRACT OR GRANT NUMBER(s) N00014-80-C-0242
9. PERFORMING ORGANIZATION NAME AND ADDRESS Center for Cybernetic Studies, UT Austin Austin, Texas 78712		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
11. CONTROLLING OFFICE NAME AND ADDRESS Office of Naval Research (Code 434) Washington, DC		12. REPORT DATE January 1981 13. NUMBER OF PAGES 12
14. MONITORING AGENCY NAME & ADDRESS(if different from Controlling Office)		15. SECURITY CLASS. (of this report) Unclassified 15a. DECLASSIFICATION/DOWNGRADING SCHEDULE
16. DISTRIBUTION STATEMENT (of this Report) This document has been approved for public release and sale; its distribution is unlimited.		
17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)		
18. SUPPLEMENTARY NOTES		
19. KEY WORDS (Continue on reverse side if necessary and identify by block number)		
20. ABSTRACT (Continue on reverse side if necessary and identify by block number) The increased importance of designing and implementing algorithms to solve particular management problems has created the need for more robust test problem generators that can match the overall structure and parameter values of these problems. Of particular interest are management problems that can be modeled using a network structure. This paper discussed the design of a system for generating network-based mathematical programming test problems that conform to user-supplied structural & parameter characteristics.		

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE
S/N 0102-014-6601

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)